

Отчёта по лабораторной работе №6

Арифметические операции в NASM.

Барето Вилиан Мануел

Содержание

1	Цель работы	4
2	Задание	5
3	Выполнение лабораторной работы	6
3.1	Символьные и численные данные в NASM	6
3.2	Выполнение арифметических операций в NASM	11
3.3	Ответы на вопросы по программе	13
3.4	Задание для самостоятельной работы	14
4	Выводы	16

Список иллюстраций

3.1	Создаем каталог с помощью команды <code>mkdir</code> и файл с помощью команды <code>touch</code>	6
3.2	Заполняем файл	7
3.3	Запускаем файл и смотрим на его работу	7
3.4	Изменяем файл	8
3.5	Запускаем файл и смотрим на его работу	8
3.6	Создаем файл	8
3.7	Заполняем файл	9
3.8	Смотрим на работу программы	9
3.9	Изменяем файл	9
3.10	Смотрим на работу программы	10
3.11	Изменяем файл	10
3.12	Смотрим на работу программы	10
3.13	Создаем файл	11
3.14	Заполняем файл	11
3.15	Смотрим на результат работы программы	11
3.16	Редактируем файл	12
3.17	Смотрим на результат работы программы	12
3.18	Создаем файл	12
3.19	Заполняем файл	13
3.20	Проверяем результат работы программы	13
3.21	Создаем файл	14
3.22	Заполняем файл	15
3.23	Проверяем работу программы	15
3.24	Проверяем работу программы	15

1 Цель работы

Освоить арифметических инструкций языка ассемблера NASM и написать программы для вычисления арифметических выражений с неизвестной.

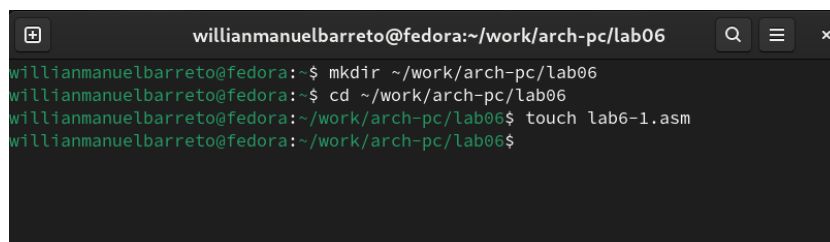
2 Задание

Написать программы для решения выражений.

3 Выполнение лабораторной работы

3.1 Символьные и численные данные в NASM

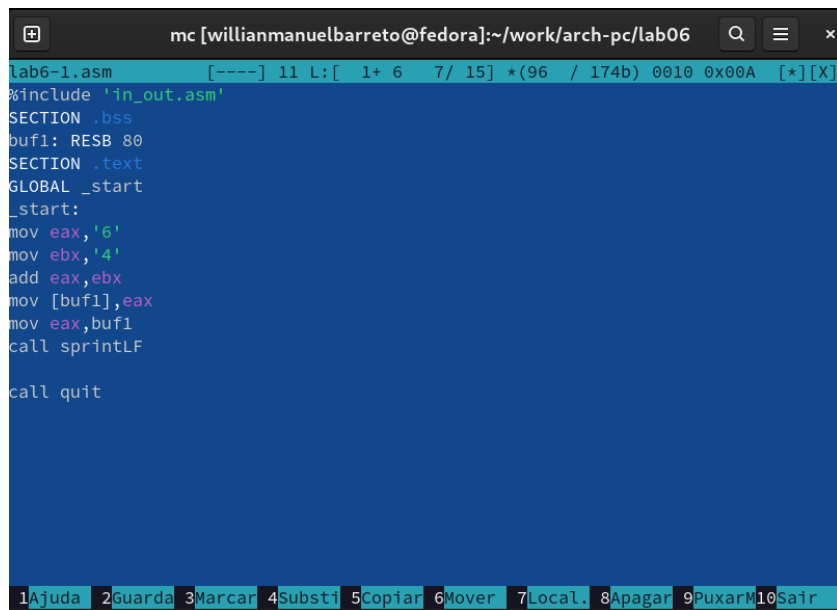
Создаем каталог для программ ЛБ6, и в нем создаем файл (рис. fig. 3.1).

A terminal window with a dark background and light green text. The title bar shows the user 'willianmanuelbarreto@fedora' and the current directory '~/work/arch-pc/lab06'. The terminal contains four lines of commands and their outputs: 1. 'mkdir ~/work/arch-pc/lab06' followed by a new prompt. 2. 'cd ~/work/arch-pc/lab06' followed by a new prompt. 3. 'touch lab6-1.asm' followed by a new prompt. 4. A final prompt without a command.

```
willianmanuelbarreto@fedora:~$ mkdir ~/work/arch-pc/lab06
willianmanuelbarreto@fedora:~$ cd ~/work/arch-pc/lab06
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ touch lab6-1.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.1: Создаем каталог с помощью команды `mkdir` и файл с помощью команды `touch`

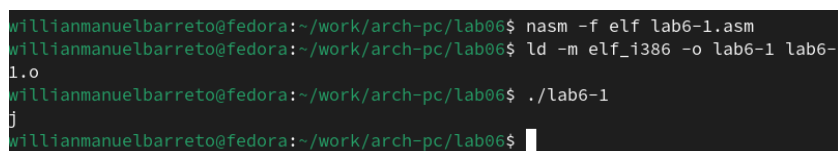
Открываем файл в Midnight Commander и заполняем его в соответствии с листингом 6.1 (рис. fig. 3.2).



```
lab6-1.asm  [----] 11 L: [ 1+ 6 7/ 15] *(96 / 174b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .bss
buf1: RESB 80
SECTION .text
GLOBAL _start
_start:
mov eax, '6'
mov ebx, '4'
add eax, ebx
mov [buf1], eax
mov eax, buf1
call printf
call quit
```

Рис. 3.2: Заполняем файл

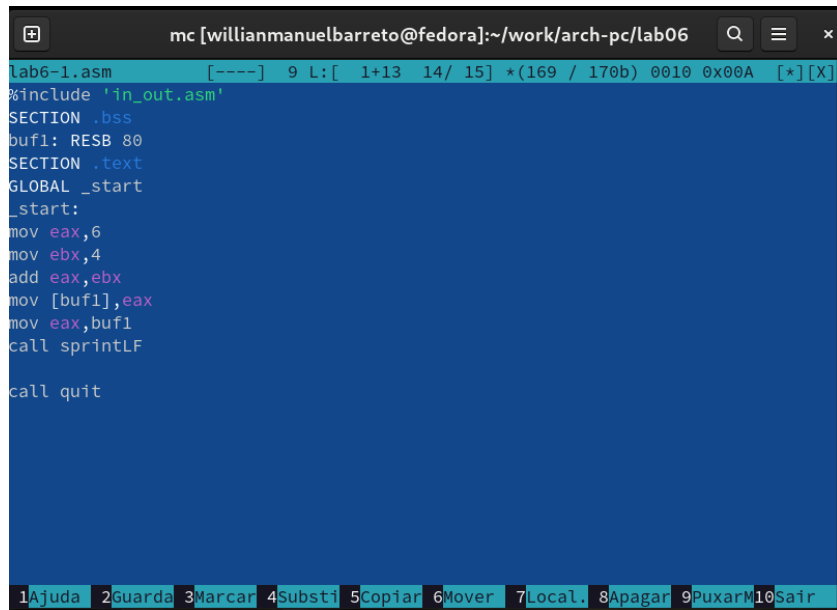
Создаем исполняемый файл и запускаем его (рис. fig. 3.3).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-1
j
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.3: Запускаем файл и смотрим на его работу

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. fig. 3.4).

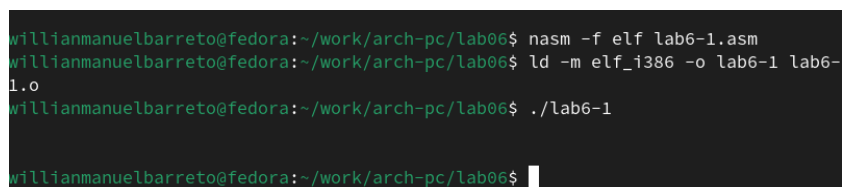


The screenshot shows a text editor window titled 'mc [willianmanuelbarreto@fedora]:~/work/arch-pc/lab06'. The editor displays the contents of 'lab6-1.asm'. The code includes a header comment, an include directive for 'in_out.asm', and assembly instructions for memory allocation and printing. The bottom of the window features a menu bar with options: 1Ajuda, 2Guarda, 3Marcar, 4Substi, 5Copiar, 6Mover, 7Local., 8Apagar, 9PuxarM, 10Sair.

```
lab6-1.asm      [----]  9 L: [ 1+13 14/ 15] *(169 / 170b) 0010 0x00A [*][X]  
%include 'in_out.asm'  
SECTION .bss  
buf1: RESB 80  
SECTION .text  
GLOBAL _start  
_start:  
mov eax,6  
mov ebx,4  
add eax,ebx  
mov [buf1],eax  
mov eax,buf1  
call sprintLF  
  
call quit
```

Рис. 3.4: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. fig. 3.5).

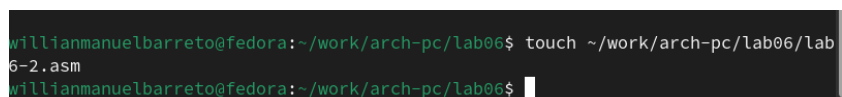


The screenshot shows a terminal window with the following commands and output:

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-1.asm  
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-1 lab6-1.o  
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-1  
  
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.5: Запускаем файл и смотрим на его работу

Создаем новый файл в каталоге (рис. fig. 3.6).

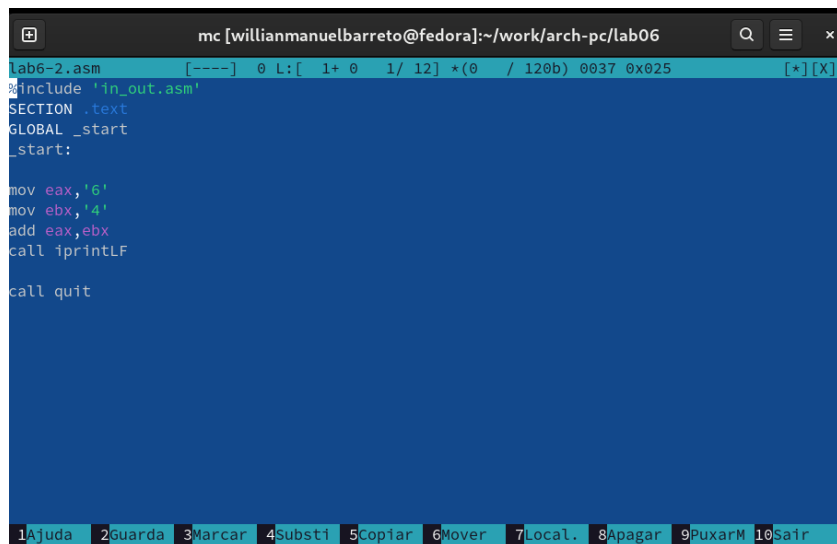


The screenshot shows a terminal window with the following commands and output:

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-2.asm  
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.6: Создаем файл

Заполняем файл в соответствии с листингом 6.2 (рис. fig. 3.7).



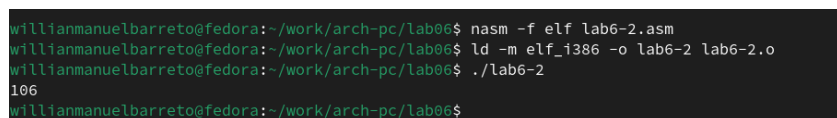
```
lab6-2.asm  [----]  0 L: [ 1+ 0 1/ 12] *(0 / 120b) 0037 0x025  [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax,'6'
mov ebx,'4'
add eax,ebx
call iprintLF

call quit
```

Рис. 3.7: Заполняем файл

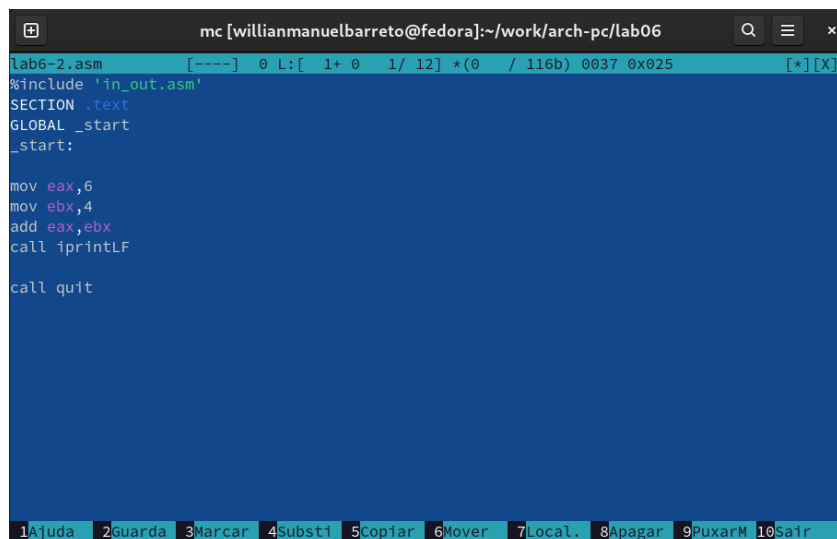
Создаем исполняемый файл и запускаем его (рис. fig. 3.8).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-2
106
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.8: Смотрим на работу программы

Снова открываем файл для редактирования и убираем кавычки с числовых значений (рис. fig. 3.9).



```
lab6-2.asm  [----]  0 L: [ 1+ 0 1/ 12] *(0 / 116b) 0037 0x025  [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprintLF

call quit
```

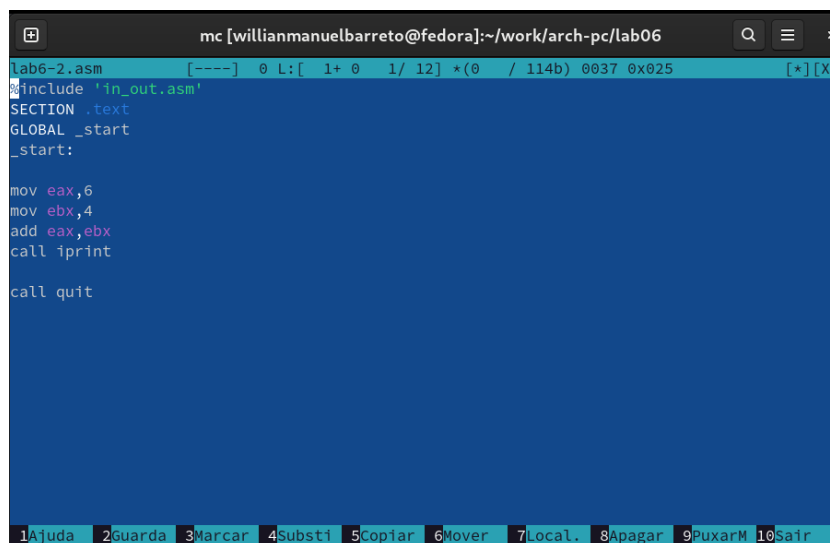
Рис. 3.9: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. fig. 3.10).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-2
10
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.10: Смотрим на работу программы

Снова открываем файл для редактирования и меняем `iprintLF` на `iprint` (рис. fig. 3.11).



```
lab6-2.asm [----] 0 L: [ 1+ 0 1/ 12] *(0 / 114b) 0037 0x025 [*][X]
#include 'in_out.asm'
SECTION .text
GLOBAL _start
_start:

mov eax,6
mov ebx,4
add eax,ebx
call iprint

call quit
```

Рис. 3.11: Изменяем файл

Создаем исполняемый файл и запускаем его (рис. fig. 3.12).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-2.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-2 lab6-2.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-2
10
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.12: Смотрим на работу программы

Вывод функций `iprintLF` и `iprint` отличаются только тем, что `LF` переносит на новую строку.

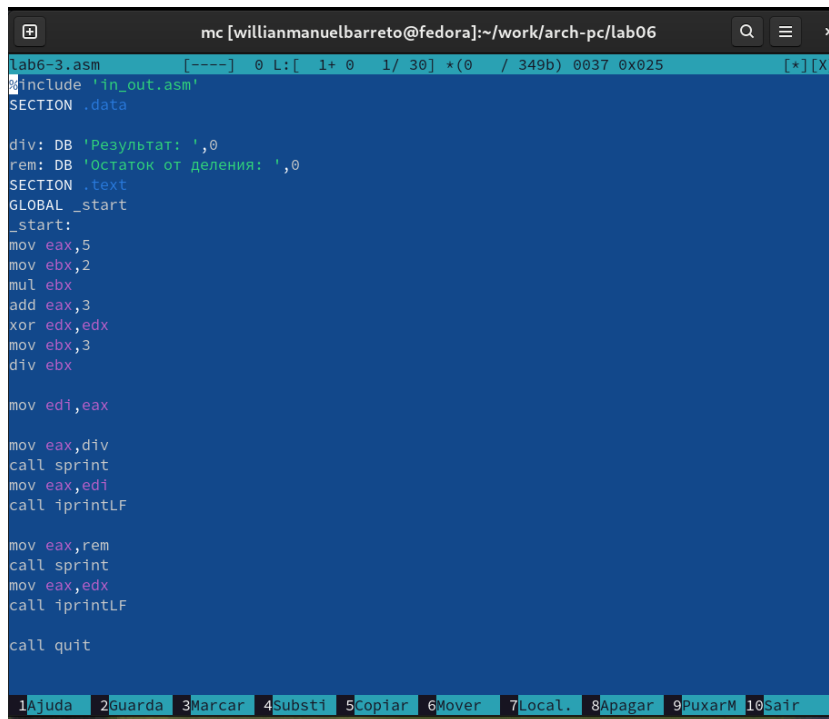
3.2 Выполнение арифметических операций в NASM

Создаем новый файл в каталоге (рис. fig. 3.13).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-3.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.13: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.3 (рис. fig. 3.14).



```
mc [willianmanuelbarreto@fedora:~/work/arch-pc/lab06]
lab6-3.asm [----] 0 L: [ 1+ 0 1/ 30] *(0 / 349b) 0037 0x025 [*][X]
#include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,5
mov ebx,2
mul ebx
add eax,3
xor edx,edx
mov ebx,3
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit
```

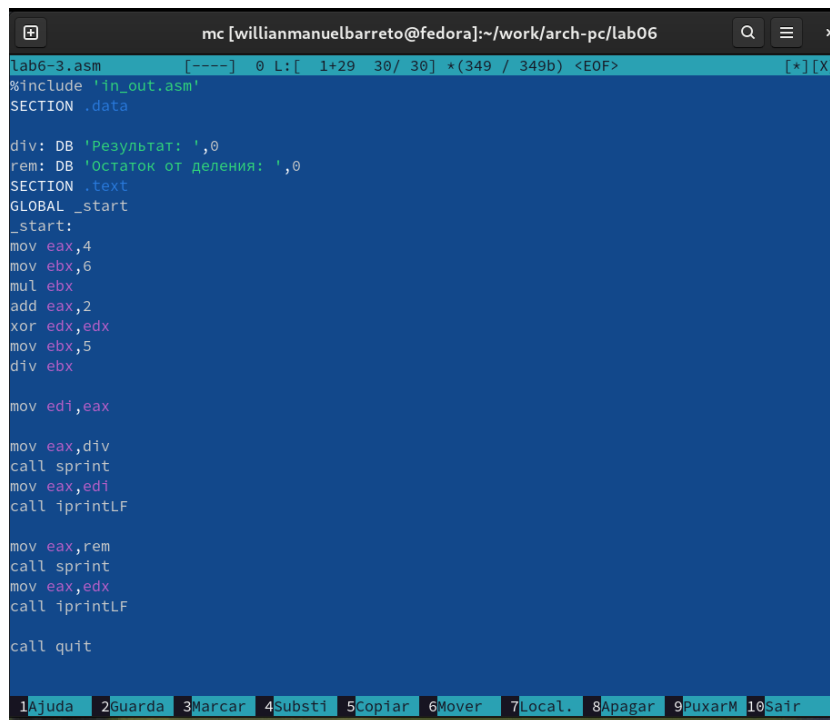
Рис. 3.14: Заполняем файл

Создаем исполняемый файл и запускаем его (рис. fig. 3.15).

```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 4
Остаток от деления: 1
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.15: Смотрим на результат работы программы

Открываем файл и редактируем его для вычисления выражения $f(x) = (4x + 2)/5$ (рис. fig. 3.16).



```
mc [willianmanuelbarreto@fedora]:~/work/arch-pc/lab06
lab6-3.asm [----] 0 L: [ 1+29 30/ 30] *(349 / 349b) <EOF> [*][X]
#include 'in_out.asm'
SECTION .data
div: DB 'Результат: ',0
rem: DB 'Остаток от деления: ',0
SECTION .text
GLOBAL _start
_start:
mov eax,4
mov ebx,6
mul ebx
add eax,2
xor edx,edx
mov ebx,5
div ebx

mov edi,eax

mov eax,div
call sprint
mov eax,edi
call iprintLF

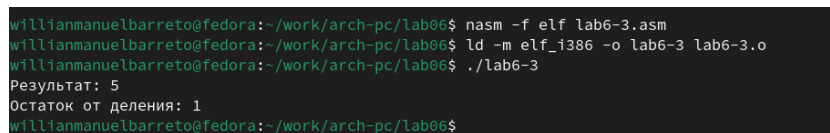
mov eax,rem
call sprint
mov eax,edx
call iprintLF

call quit

1Ajuda 2Guarda 3Marcar 4Substi 5Copiar 6Mover 7Local. 8Apagar 9PuxarM 10Sair
```

Рис. 3.16: Редактируем файл

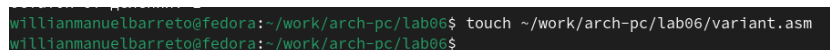
Компилируем файл и запускаем программу (рис. fig. 3.17).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-3.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-3 lab6-3.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-3
Результат: 5
Остаток от деления: 1
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.17: Смотрим на результат работы программы

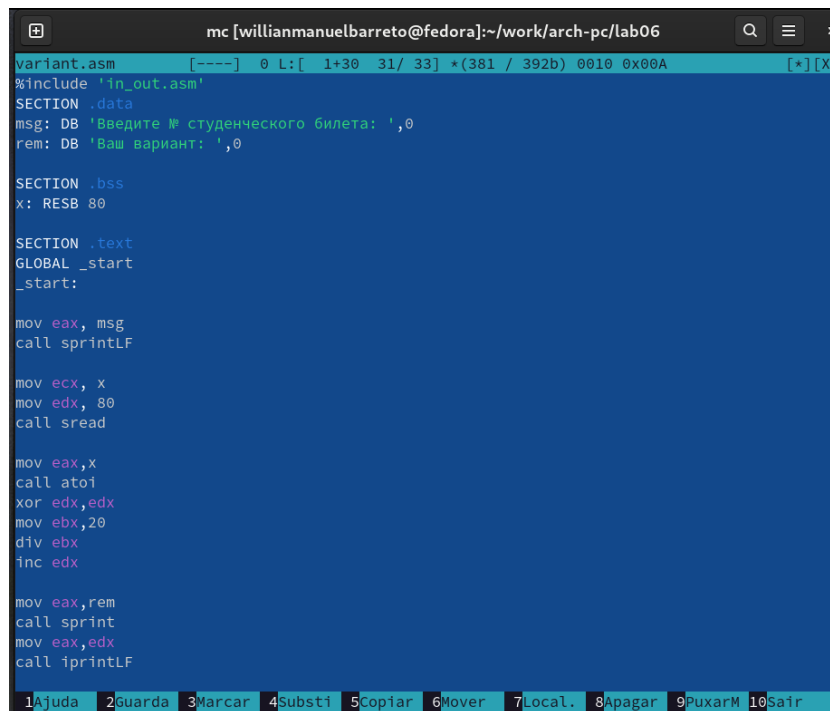
Создаем новый файл в каталоге (рис. fig. 3.18).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/variant.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.18: Создаем файл

Открываем файл и редактируем в соответствии с листингом 6.4 (рис. fig. 3.19).



```
variant.asm [-----] 0 L: [ 1+30 31/ 33] *(381 / 392b) 0010 0x00A [*][X]
#include 'in_out.asm'
SECTION .data
msg: DB 'Введите № студенческого билета: ',0
rem: DB 'Ваш вариант: ',0

SECTION .bss
x: RESB 80

SECTION .text
GLOBAL _start
_start:

mov eax, msg
call sprintf

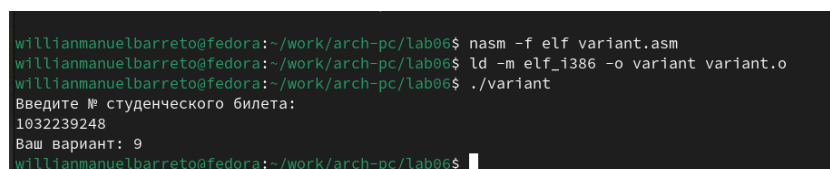
mov ecx, x
mov edx, 80
call sread

mov eax, x
call atoi
xor edx, edx
mov ebx, 20
div ebx
inc edx

mov eax, rem
call sprintf
mov eax, edx
call iprintfLF
```

Рис. 3.19: Заполняем файл

Компилируем файл и запускаем его (рис. fig. 3.20).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf variant.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o variant variant.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./variant
Введите № студенческого билета:
1032239248
Ваш вариант: 9
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.20: Проверяем результат работы программы

3.3 Ответы на вопросы по программе

1. Строка “mov eax,rem” и строка “call sprint” отвечают за вывод на экран сообщения ‘Ваш вариант:’.
2. Эти инструкции используются для чтения строки с вводом данных от пользователя. Начальный адрес строки сохраняется в регистре ecx, а количество символов в строке (максимальное количество символов, которое может

быть считано) сохраняется в регистре `edx`. Затем вызывается процедура `sread`, которая выполняет чтение строки.

3. Инструкция `call atoi` используется для преобразования строки в целое число. Она принимает адрес строки в регистре `eax` и возвращает полученное число в регистре `eax`.
4. Строка `xor edx,edx` обнуляет регистр `edx` перед выполнением деления. Строка `mov ebx,20` загружает значение 20 в регистр `ebx`. Строка `div ebx` выполняет деление регистра `eax` на значение регистра `ebx` с сохранением частного в регистре `eax` и остатка в регистре `edx`.
5. Остаток от деления записывается в регистр `edx`.
6. Инструкция `inc edx` используется для увеличения значения в регистре `edx` на 1. В данном случае, она увеличивает остаток от деления на 1.
7. Строка `mov eax,edx` передает значение остатка от деления в регистр `eax`. Строка `call iprintLF` вызывает процедуру `iprintLF` для вывода значения на экран вместе с переводом строки.

3.4 Задание для самостоятельной работы

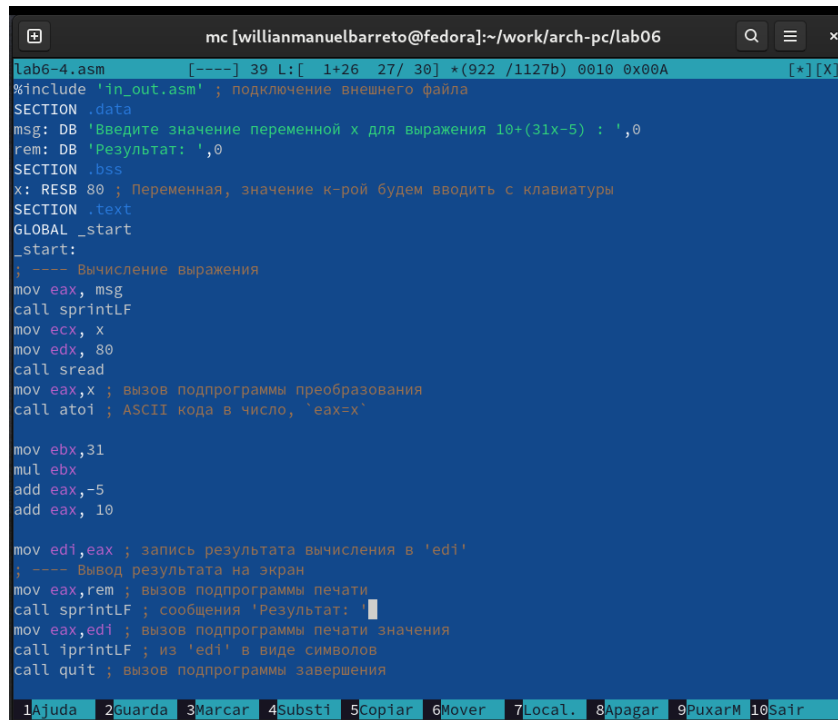
Создаем новый файл в каталоге (рис. fig. 3.21).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ touch ~/work/arch-pc/lab06/lab6-4.asm
```

Рис. 3.21: Создаем файл

Открываем его и заполняем, чтобы решалось выражение $f(x)=10 + (31x - 5)$ (рис. fig. 3.22).



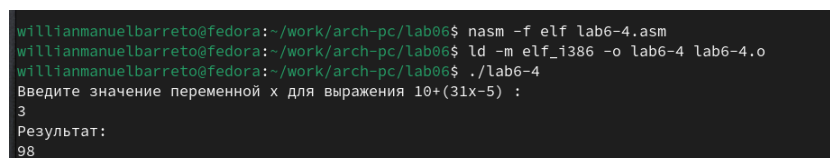
```
lab6-4.asm [----] 39 L: [ 1+26 27/ 30] *(922 /1127b) 0010 0x00A [*][X]
#include 'in_out.asm' ; подключение внешнего файла
SECTION .data
msg: DB 'Введите значение переменной x для выражения 10+(31x-5) : ',0
rem: DB 'Результат: ',0
SECTION .bss
x: RESB 80 ; Переменная, значение к-рой будем вводить с клавиатуры
SECTION .text
GLOBAL _start
_start:
; ---- Вычисление выражения
mov eax, msg
call sprintf
mov ecx, x
mov edx, 80
call sread
mov eax, x ; вызов подпрограммы преобразования
call atoi ; ASCII кода в число, 'eax=x'

mov ebx, 31
mul ebx
add eax, -5
add eax, 10

mov edi, eax ; запись результата вычисления в 'edi'
; ---- Вывод результата на экран
mov eax, rem ; вызов подпрограммы печати
call sprintf ; сообщения 'Результат: '
mov eax, edi ; вызов подпрограммы печати значения
call iprintf ; из 'edi' в виде символов
call quit ; вызов подпрограммы завершения
```

Рис. 3.22: Заполняем файл

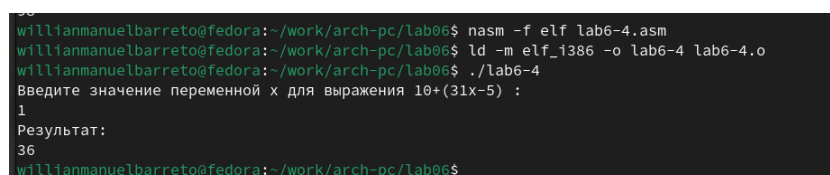
Компилируем программу и проверяем для $x=3$ (рис. fig. 3.23).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите значение переменной x для выражения 10+(31x-5) :
3
Результат:
98
```

Рис. 3.23: Проверяем работу программы

Компилируем программу и проверяем для $x=1$ (рис. fig. 3.24).



```
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ nasm -f elf lab6-4.asm
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ld -m elf_i386 -o lab6-4 lab6-4.o
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$ ./lab6-4
Введите значение переменной x для выражения 10+(31x-5) :
1
Результат:
36
willianmanuelbarreto@fedora:~/work/arch-pc/lab06$
```

Рис. 3.24: Проверяем работу программы

4 Выводы

Мы приобрели навыки создания исполнительных файлов для решения выражений и освоили арифметические инструкции в NASM.